

Distributed & Mobile-Systems

Auftrag für Leistungsnachweis

Fach	Distributed & Mobile-Systems	Bearbeitungszeit	ca 7h
Titel	Weather Klasse		
Studiengruppe	BWI-A19	Anzahl Aufgaben	Einzelarbeit
Abgabe-Datum	23.11.21 / 17:00	Bewertung	<p>Geprüft wird der Code nach vorgegebenen funktionalen und Qualitäts-Kriterien.</p> <p>Anhand eines Fachgespräches wird überprüft, ob der Student den Code und die darunter liegenden theoretischen Grundlagen verstanden hat. (Gewicht: 100%)</p>

Erlaubte Hilfsmittel	<ul style="list-style-type: none"> Alles inkl. Google, PyCharm, ... ausser 1:1 Abschreiben oder Kopieren ohne Deklaration als Kommentar im Code.
Auftrag	<ul style="list-style-type: none"> Design und Implementation einer allgemeinen Wetterstations-Klasse Entwickeln einer CLI Test-Applikation, welche diese Klasse nutzt und zum Testen der Klasse verwendet werden kann.

Grobbeschreibung

Um einen Web-Service in Python zu nutzen, wird oft vom Service-Provider eine Klasse (Module) zur Verfügung gestellt. Diese Klasse kann der Entwickler verwenden, um Objekte zu kreieren und in der seine eigene Applikation nach OO-Richtlinien zu designen und zu implementieren.

Entwickeln Sie eine allgemeine WeatherStation-Class, welche für verschiedenste Web-Services für Wetterdienste verwendet werden kann. Ein konkreter Wetterdienst (z.B. OpenWeather) muss implementiert sein. Dabei ist das API des Wetterdienstes nur in der Implementation sichtbar und somit für den Verwender der Klasse (Applikation) nicht.

Für diese Klasse entwickeln Sie eine CLI-Applikation, mit welcher die Klasse getestet werden kann.

Für diese Aufgabenstellung „packen“ Sie alles in ein einziges Python File und fügen Kommentare und Docstrings mit Ihren Überlegungen dazu.

Functional and Quality-Requirements

- Lauffähige Test-Applikation rechtzeitig abgegeben
- Alles in **einem** File (ausnahmsweise für diese Aufgabenstellung)
- Filename: Vorname_Nachname_A19_DS.py (z.B. Rea_Vogel_A19_DS.py)
- Cleancode Regeln berücksichtigt
- Wetter-Klasse vorhanden
 - Design dokumentiert (z.B. mit Class-Diagramm), publiziert und in Klasse verlinkt
 - Design stimmt mit Implementation überein
 - `__str__` and `__eq__` implemented
 - Initializer overloaded mit Default-Values
 - Klares, verständliches und einfaches Interface
 - Sichtbarkeitsregeln beachtet
 - OO-Design und Encapsulation nachvollziehbar
 - Methoden-Argumente haben sinnvolle Default-Werte
 - Methoden haben allgemein strukturierte Return-Values
 - Klasse greift auf Wetterdienst-Daten zu
- Application
 - Eigene Appld gelöst
 - User-Interaktivität vorhanden
- Test
 - Test-Fälle dokumentiert
 - Test-Statistik vorhanden
 - Test-Abdeckung genügend
 - Positive wie negative Testfälle implementiert
 - Test-Driven approach erkennbar

Example Wetterdienst

Beispiel für REST-Call (OpenWeather)

OpenWeather Web-App: <https://openweathermap.org/city/7287397>

API doc and subscription: <https://openweathermap.org/api>

Get a free API-Key for Current weather data (Limited to 60 calls/minute or 1Mio calls/month →
alle 10" request = 267840)

```
# -----  
# Name: WeatherLogger.py  
#  
# Description: Polling REST Service and write values to console  
#  
# Autor: Walter Rothlin  
#  
# History:  
# 03-Dec-2020    Walter Rothlin    Initial Version ()  
# 10-Oct-2021    Walter Rothlin    Adapted for BWI-A20  
# -----  
  
import requests  
import json  
import time  
  
pollingTime = float(input("Polling-Time [s]:"))  
serviceURL = "https://api.openweathermap.org/data/2.5/weather"  
appId = "144747fd356c86e7926ca91ce78ce170"  
while True:  
    responseStr = requests.get(serviceURL + "?q=Uster&units=metric&lang=de&appid=" + ap-  
pId)  
    jsonResponse = json.loads(responseStr.text)  
  
    temp = jsonResponse['main']['temp']  
    pressure = jsonResponse['main']['pressure']  
    humidity = jsonResponse['main']['humidity']  
    lon = jsonResponse['coord']['lon']  
    lat = jsonResponse['coord']['lat']  
    cloud = jsonResponse['weather'][0]['description']  
  
    print(temp, pressure, humidity, lon, lat, cloud)  
    time.sleep(pollingTime)  
    time.sleep(pollingTime)
```

JSON Response to

<http://api.openweathermap.org/data/2.5/weather?q=Wangen%20SZ&appid=3836093dde650898eb014e6f27304646>

```

▼ coord:
  lon:      8.89
  lat:      47.19
▼ weather:
  ▼ 0:
    id:      800
    main:    "Clear"
    description: "clear sky"
    icon:    "01n"
    base:    "stations"
▼ main:
  temp:      275.35
  feels_like: 272.31
  temp_min:  274.82
  temp_max:  275.93
  pressure:  1022
  humidity:  85
  visibility: 10000
▼ wind:
  speed:     1.5
  deg:       151
▼ clouds:
  all:       0
  dt:        1606237231
▼ sys:
  type:      3
  id:        2006037
  country:   "CH"
  sunrise:   1606200100
  sunset:    1606232470
  timezone:  3600
  id:        2658054
  name:      "Wangen"
  cod:       200

```